Melanoma Detection with Smartphone-based Medical Imaging

Taha Zaman Khan Khattak, Maia Ghionea, Gergely Takács, Vincenzo Emanuele Piras, Nielson Völk

https://github.com/Niels04/2025-FYP-Final-Group-Eastern-Mud-Turtle.git

Contents

1 Introduction

Early detection and diagnosis of skin cancer is critical to its cure, however, not all patients are fortunate enough to visit a doctor, or our current healthcare infrastructure is simply not capable enough to check every skin lesion for melanoma. In the last few years, computer-aided diagnosis systems have been proposed for skin lesion analysis, yet, a major downside of most systems is the usage of dermoscopic images. Unfortunately, such imagery is not universally available, and the cost involved is sometimes too high. In this project, we make an attempt to find a solution to this problem using what almost everyone has access to in today's world, a smartphone.

1.1 Melanoma Diagnosis

Doctors and dermatologists use various methods to detect melanoma skin cancer. According to (ACS, 2023), the first step a doctor takes in melanoma diagnosis is inquiring when the lesion first appeared, if it has changed in size, itched or bled. The doctor may also ask about risk factors, such as family history of cancer, sunburns, etc. If a doctor suspects melanoma skin cancer, the patient will be referred to a dermatologist.

A dermatologist will then confirm if it is melanoma using different methods, the main one being skin biopsy. Here a part or the entire top layer of the lesion is removed and sent to a laboratory for microscopic inspection. In the laboratory, a pathologist will look for cancerous cells.

As mentioned above, not everyone has access to these resources, or medical experts simply do not have enough time to inspect each and every lesion. Thus, experts have developed the ABCDE rule for melanoma detection.

1.2 ABCDE Technique

The ABCDE technique for melanoma detection stands for:

A - Asymmetry
How asymmetric is the shape of the lesion? A more asymmetrical or uneven shape could indicate melanoma skin cancer.

B - Border
How compact is the border? A less compact border i.e irregular border indicates melanoma skin cancer.

C - Color
How many different color clusters are present? A wide range of colors may indicate melanoma.

D - Diameter
How long is the lesion from one end to the other? A lesion with a diameter longer than 5-6mm may be an indication of melanoma.

E - Evolving
Has the lesion increased in size? An evolving lesion is also an indication of melanoma.

Further explanation of these features and their implementations in our project can be found in Section 3.

Our base model uses asymmetry, border and color as features, whereas our extended model also make use of the features listed below.

BV - Blue Veil
How much of the lesion is blue/purple in color? A higher ratio of blue and/or purple structures could be an indication of melanoma.

CH - Connected Components
How many connected patches is the lesion composed of?

S - Snowflake
Are there any bright white spots in the lesion?

2 Data

The data set has been provided to us by the Federal University of Espírito Santo, Brazil (Pacheco et al., 2020). It consists of 2298 images of skin lesions captured using smartphones.

2.1 Data Format

The data set is composed of three folders and an associated *metadata.csv* file containing relevant information on said images.
In particular, the most notable columns in the metadata file are:

- The *patient_id* column, which assigns to each patient a unique number (e.g. PAT_123),
- The *lesion_id* column, which assigns to each individual lesion a unique number (e.g. 456),
- The *diagnostic* column, which stores the diagnosis of the image as a three letter string.

The first two are fundamental for identifying and referencing images, but also for handling cases of multiple instances of the same lesion to then

correctly evaluate the model.
This aspect is analyzed in more depth in Section 4.2.
The last column is, quite obviously, needed to compare the predictions with the real labels of the lesions while training the model.
Other information contained in the metadata file refers to personal data (such as age, gender, health history and daily habits), lesion data (where it is located, if it is expanding, if it hurts or bleeds, ...) and data regarding living conditions (presence of sewage systems, access to piped water).
None of these attributes are used in the development and training of the model, which is entirely based on the lesion image itself.
In summary, our method uses the file *metadata.csv* exclusively to map an image to a patient and lesion number, and to evaluate the predictions made by the model.

Together with the lesion images, a folder of mask images made by previous students was included. The vast majority of the data points follow a specific naming convention. For the image *'lesion.png'*, the corresponding mask will hold the name of *'lesion_mask.png'*. Concretely, the patient id merged with the lesion id, followed by random digits, constitutes the name of lesion images. An example of a properly named lesion image and corresponding mask image is:
PAT_123_456_999.png,
PAT_123_456_999_mask.png

2.2 Data Cleaning

The data set provided presents some minor issues. Because of that, some functions are to be adjusted in order for these anomalies to be properly dealt with.

The main instance of this is the `ImageDataLoader` class,
in which two checks are present to ensure that only valid image - mask pairs are loaded.
As mentioned in Section 2.1, a naming convention of the data set makes it possible to link a lesion image to the corresponding mask image. Unfortunately, a few dozens data points, precisely 48, deviate from this scheme, making it impossible to retrieve the mask image based off the name of the lesion image, and vice-versa.

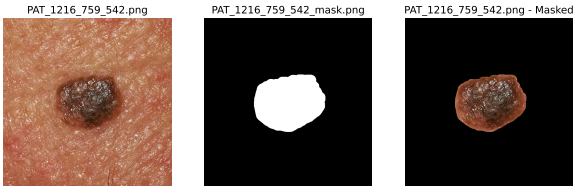

Figure 1: An example of consistent naming, which enables proper image loading and masking.

So, in similar cases, instead of trying to load a non-existent image, the class skips to the next iteration of the loading loop, losing the mask - image pair but avoiding errors.

The second check regards fully black masks. If the program loads a mask that, after binary conversion, only contains zeros, it discards the mask - image pair and skips to the next iteration. This prevents various parts of the code from outputting unexpected values, or errors, since the model expects exclusively images that contain a lesion (as is the case in the original data set).
There are 3 instances of masks that, when binarized, contain exclusively zeros.

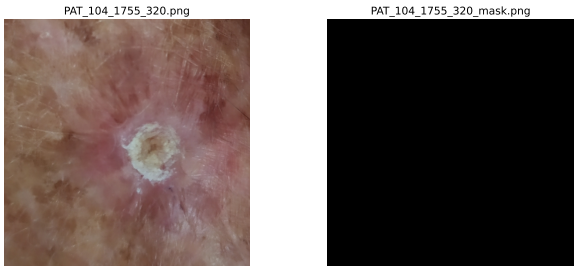

Figure 2: One of the instances of a totally black mask.

In summary, the dataset came with few inconsistencies and criticalities, but slight targeted modifications to the code in specific sections make it possible to process the data points with only minor losses (51 mask - image pairs are lost, out of 2151 total).

3 Feature Extraction

When testing for melanoma, the most well-known method of identification is through the ABCDE features. This acronym was designed with the aim of making this common cancer type easily identifiable by the public and clinicians through its memorable format. This section dissects the implemen-

tation and logic of the code for each of these feature extractors and states the correlation between our numerical outputs and real-life clinical observations. Additionally, our own features are also explained below, such as the blue veil, connected components, and the snowflake features.

3.1 Asymmetry (A)

- This feature returns a value between 0 and 1, detailing the symmetry score of the lesion, as this hints at the possible presence of melanoma.

- The process involves cropping the image to the contour of the lesion and splitting this into quarters, which are compared to each other pairwise using xor logic. This comparison calculates an asymmetry score between these pairs, after which the image is rotated a set number of times. For each rotation the asymmetry is calculated, of which the mean is returned.

3.2 Border (B)

- The border feature rates the degree of the lesion's compactness, by checking how close to a circle it is. The result is a value in the range [0, 1], where 0 represents no compactness, and 1 means maximum compactness, a perfect circle.

- By cropping the image to the lesion's proportion, adding a tiny border to the mask as padding, and adjusting the image resolution (this is necessary for comparing the scores among different medical cases), the lesion image is prepared for analysis.

- Our focal point for this feature is the border irregularity, which, in broader terms, equals to the compactness measure of the mask. To execute such analysis we first obtain the lesions perimeter, after which the compactness formula is applied.

3.3 Color (C)

- This feature uses three chained functions to count the number of colors present inside of the lesion. While a melanoma can be brown, a high count of hues can be an indicator of cancerous presence, especially in lesions reaching 4 to 6 different colors.

- First, an ordered list of all RGB values of the colors is created using `get_multicolor_rate`, where the first color is the most prevalent. This list is reduced based on a set threshold, to only contain the colors differentiable by the human eye. Thus, the base color is be the first in the list and the rest represent various lesion colors. The length of this list is returned.

3.4 Hair Feature

The Hair feature function extracts and quantifies the hair present in a given image. It is based heavily on the method proposed by (Mostame, 2023).
The process used is as follows.

- The feature computes the ratio of hair pixels to total image pixels, then translates it into one of three labels 0 (little to no hair), 1 (moderate amount of hair), 2 (substantial amount of hair).

- First, it blurs the image, then it enhances dark structures, filtering out lighter regions.

- Starting from these darker edges, it traces lines to connect them. The resulting mask, composed of these lines, is then dilated.

- Finally, it computes the number of pixels in the mask, considered hair, and divides it by the total number of pixels, obtaining a ratio with a value between 0 and 1. Following set thresholds, whose optimization process is described below, the ratio gets transformed into a discrete label.

The function was tested on manually annotated images, not present in the data set used for the model described in this paper.
In particular, 200 manually annotated images are used to optimize the aforementioned thresholds by iterating through all possible threshold values. The obtained results can be consulted in Figure 3, where the necessity to blur the image is justified by higher accuracy scores.

100 other manually annotated images are used to test the accuracy of the function with the newly set thresholds. The final score is 78.79%, a notable outcome.

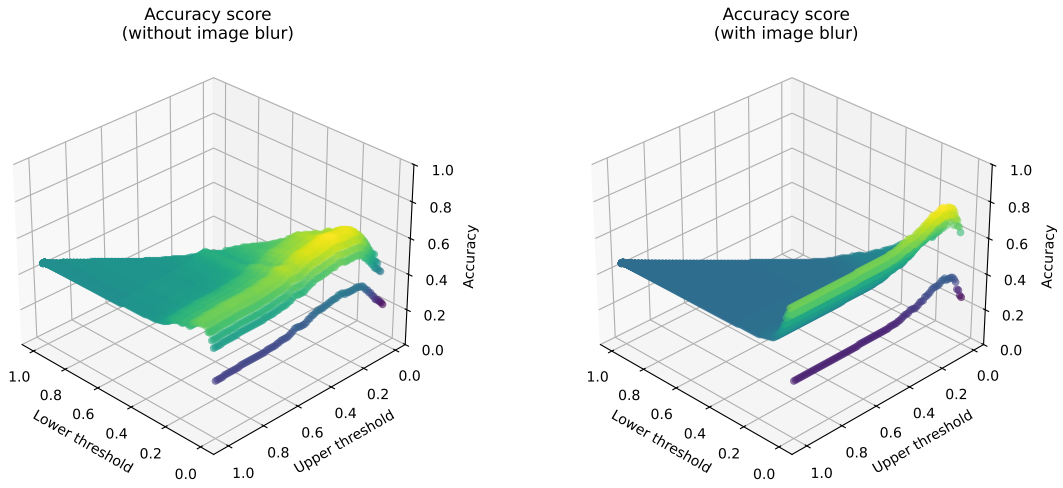

Figure 3: Plot of hair-rating function accuracy based on threshold values, for blurred and non-blurred images.

Unfortunately, the data set used to develop the model contains a multitude of images portraying dry, rough or chapped skin, characteristic not present in such quantities in the annotated images. Therefore, the function cannot distinguish reliably between certain strand-like skin types and hair in some cases [Figure 4], but captures the general hair pattern quite well in others [Figure 5].
This skin related issue is the main reason why the model does not concretely inpaint the supposed hair picked up by the function, but stops at labeling it: we deem not worth to alter the images when there is a tendency to wrongly identify skin segments as hair.

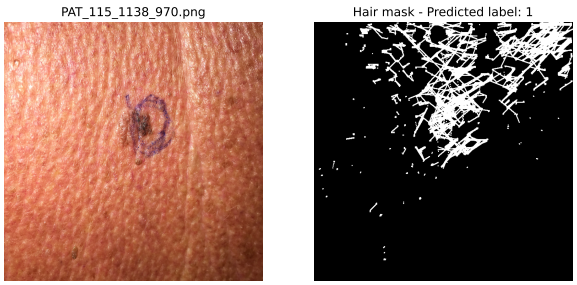

Figure 4: Case of confusion between dry skin and hair, leading to incorrect labeling.

3.5 Blue Veil (BV)

- This feature computes the percentage of the lesion that consists of blue / purple-ish pixels, the so called blue veils. It outputs a value between 0 and 1, where a 0 indicates total absence of blue veils, and a 1 indicates that the lesion is completely covered in blue veils. The presence of such hues in a lesion are impactful on the diagnosis, therefore we take account of it.

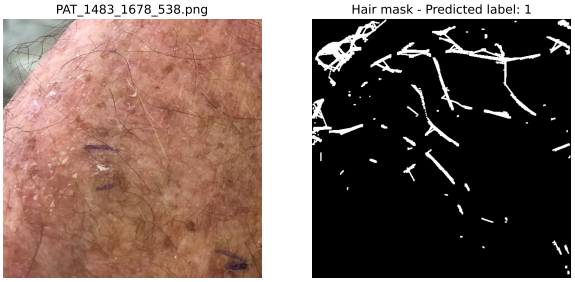

Figure 5: Case of correct highlighting of general hair pattern, leading to reasonable labeling.

- Due to previous issues, all masks are resized to fit their corresponding lesion image. All non-blue / purple-ish pixels are isolated using a second mask, after which the remainder of pixels are counted and the ratio to the total pixel count is calculated.

3.6 Connected Components(Ch)

- The CH feature returns the number of patches, which make up the lesion in the image.

- Due to noise pixels in the mask wrongly dividing the lesion into multiple parts, we dilate this mask repeatedly, with no set number of dilations until we achieve the desired

result. In each of these iterations, the number of components is counted and a mean of these values is computed and returned.

3.7 Feature Snowflake

- The snowflake feature checks if there exist white pixels in the mask region.
- The image is cropped to the dimensions of the mask and pixels are checked according to a fixed threshold. The output is binary; 1 if such pixels are present and 0 otherwise.

4 Classification

Here we briefly introduce the classifiers and hyperparameters chosen for our baseline and for our extended method. We discuss our cross-validation process used to determine these classifiers and hyperparameters. Lastly, we delve into reasons why our chosen method works well in the context of Melanoma detection from our dataset while comparing to previous techniques.

4.1 Our Method and Development Results

For the baseline and extended model, we selected the Logistic Regression with the parameter setting `class_weight="balanced"`. This automatically adjusts the weights of classes inversely proportional to their frequencies, helping to solve the strong class imbalance present in our dataset. Logistic Regression is a simple, efficient, and interpretable classifier well-suited for binary classification tasks such as melanoma detection.
The baseline Logistic Regression achieves a mean recall of 0.5397 on the training data and 0.5715 on the validation data, showing moderate sensitivity to melanoma cases. As expected, precision remained low (0.0314 train, 0.0329 validation) due to the class imbalance and the model prioritizing recall. Logistic Regression exhibited low variance in recall across multiple runs, making it more stable than some of the more complex models like Decision Trees and Voting Classifiers.

4.2 Our Cross Validation Process

In the process of finding a good classifier with appropriate hyperparameters, we assess the performance of many different such combinations. Initially we divide our dataset into a development set (80%) and a held-out test set (20%). This ensures credible test results since our classifier and we ourselves never look at the test data. Thus overfitting to the test set can neither occur by training, nor through overfitting by observer. To be able to make significant claims about the difference in performance between various approaches, we conduct a cross-validation process using 20 random grouped data splits. Data points are grouped by the column "pat_les_id", which is a combination of "patient_id" and "lesion_id" from the original dataset that allows for unique identification of each individual lesion. This ensures that images of the same lesion can never be in the training and validation data for a single split simultaneously. We use the information gained from fitting and evaluating various methods over the shuffles to visualize their performance. This includes box plots, confusion matrices Figure 6, and ROC curves. In our comparative visualizations, we include performance measured on the training part of the shuffles as well as the validation part of the shuffles. Comparing the performances on training data and validation data allows us to assess generalization capabilities of our method.

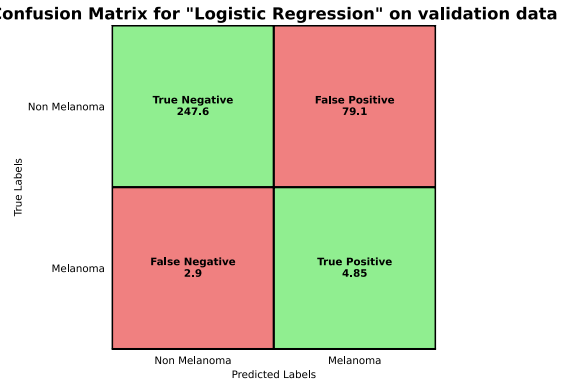

Figure 6: Confusion Matrix for Logistic Regression Classifier during development phase.

Throughout our development phase we evaluate different methods under the following metrics:

4.2.1 Recall

From a practical standpoint, we consider recall as a very important performance metric for Melanoma classification. As the ratio of "True Positives" over "True Positives + False Negatives" it shows the percentage of data points in the target class that the method correctly predicts to be in the target class.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

We believe this is crucial in the detection of lethal conditions, such as cancer, because "False

Negatives" have the high cost of a human life associated with them.

We measure recall scores over 20 random grouped shuffles to report means and variances. Visualizations such as boxplots allow us to compare methods and further investigate promising approaches Figure 7.

For reporting our final test results we use bootstrap resampling with 20 samples on our held-out test dataset. This way we can assess mean and variance of our method performance.

4.2.2 Precision

It is worth noting that, while the cost of a "False Positive" is comparatively less in the case of Melanoma classification, it is nonetheless important to keep a balance between minimizing "False Negatives"(maximizing recall) and minimizing "False Positives"(maximizing precision).

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

This is because falsely predicting Melanoma for many healthy people could cause unnecessary stress for doctors and patients alike. We judge accuracy to be an insufficient metric for our problem since it is strongly inflated by class imbalance. Only 2.33% of our extracted datapoints are Melanoma and thus our dataset shows great imbalance. Theoretically, accuracy could also serve as an indicator of increased "False Positives" but precision allows us to judge independent from class imbalance.

Precision calculation both in the development phase and the final reported results was conducted in the exact same fashion as recall calculation.

4.2.3 ROC-AUC

We also choose to include ROC-AUC as a metric for developing and evaluating our method. It has the advantage of being threshold-independent and robust to class-imbalance at the same time. Thus it can serve as a good basic performance summary for a method after which we might choose to further investigate by optimizing the threshold.

In our development phase we calculate ROC-AUC over 20 random grouped shuffles to obtain a mean performance and variation suitable for boxplot representations Figure 8. We pool all predictions and true labels over the 20 shuffles and use them to generate a combined ROC curve Figure 9.

For reporting final performance on the held-out test-data we use bootstrap resampling to obtain mean and variance of the AUC. Additionally, we generate a combined ROC-curve over 20 bootstrap samples by pooling all predictions and true labels.

4.3 Development Phase

During the development phase we explore different options for classifiers such as decision trees, random forests, k nearest neighbors(KNN), logistic regression, and voting classifiers. We measure performances with different hyperparameters and use diagnostic plots to understand reasons for performance differences.
Our dataset shows severe class imbalance with only 52 out of 2298 pictures being Melanoma. Due to this, initial tests with our classifiers show low recall performance, likely because of underfitting the Melanoma class. We mitigate this problem by using the parameter `class_weight = "balanced"` for all classifiers, except for KNN. While this improves recall, the overall performance stays low. Because of this, we fine-tune each classifier further.

K-Nearest Neighbors (KNN)

KNN is a simple classifier, that predicts the label for the samples based on the majority class among their nearest neighbors (`weights = "distance"`). It is sensitive to class imbalance, which makes its performance weaker in the context of this problem. We are able to improve its recall to 5% by using K = 1.

4.3.1 Decision Tree

Decision Trees are interpretable and fast, usable to detect non-linear relationships. However, they tend to overfit easily, especially with our imbalanced data. We obtain the the best performance at `max_depth=2` for the baseline model, suggesting that shallow trees generalize better in this setting. Deeper trees show overfitting towards the training data and perform worse on the validation set. For the extended model we measure best results with `max_depth=5`, which reflects the fact that more features are in use.

4.3.2 Random Forest

Random Forests are an ensemble method that builds multiple decision trees and averages their predictions to improve generalization. We get the

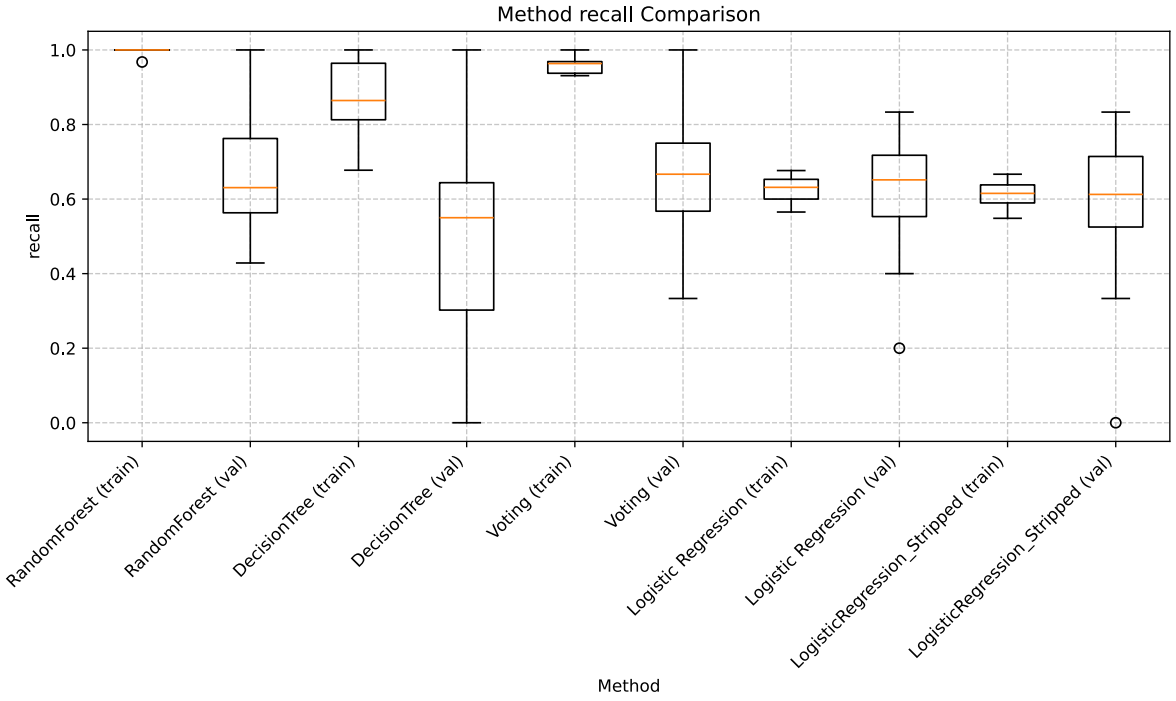

Figure 7: Boxplot of recall measurements during development phase.

best performance at `max_depth=2` for the baseline model and at `max_depth=5` for the extended model, similarly to the decision tree.

4.3.3 Logistic Regression

Logistic Regression is a linear model used for binary classification. It estimates the probability that a given input belongs to a particular class using the logistic (sigmoid) function. Learns weights by minimizing the log loss.

4.3.4 Voting Classifier

Voting Classifier combines multiple base classifiers, and predicts based on the average probabilities, also called soft voting. We have used the Decision Tree, Random Forest and the Logistic Regression for this classifier. It combines the strength of each model, it, however, strongly depends on the quality of individual models, also computation heavy.

4.4 Analysis of Development Results

We aim to come up with hypotheses as to why Logistic Regression with the chosen hyperparameters can achieve good performance in the context of this problem.

4.4.1 Previous Work and Linearity

While trying to investigate why logistic regression gives better performance than other models explored, we found that this is not the first time in CAD that logistic regression has proven to show better results than other methods in prediction tasks as shown by previous research. (Warjurkar and Ridhorkar, 2021) found logistic regression surpassing all other methods, and attaining 97.15% accuracy while investigating ML models in detection of brain tumor and Parkinson's disease.

Additionally, previous research has shown that Melanoma diagnosis using the ABCD rule can be conducted with a linear equation that assigns weights to visually quantified features of the lesion.

$$y = 1.3a + 0.1b + 0.5c + 0.5d$$

(Nachbar et al., 1994) We continue exploring this particular approach in our open question. The proposed method has a decision threshold of 4.75, i.e., the practitioner would classify any lesion exceeding $y = 4.75$ as melanocytic skin cancer. This suggests a linearity of the relationship between the "ABCD-features" and the Melanoma classification. As a result, it is reasonable to believe that

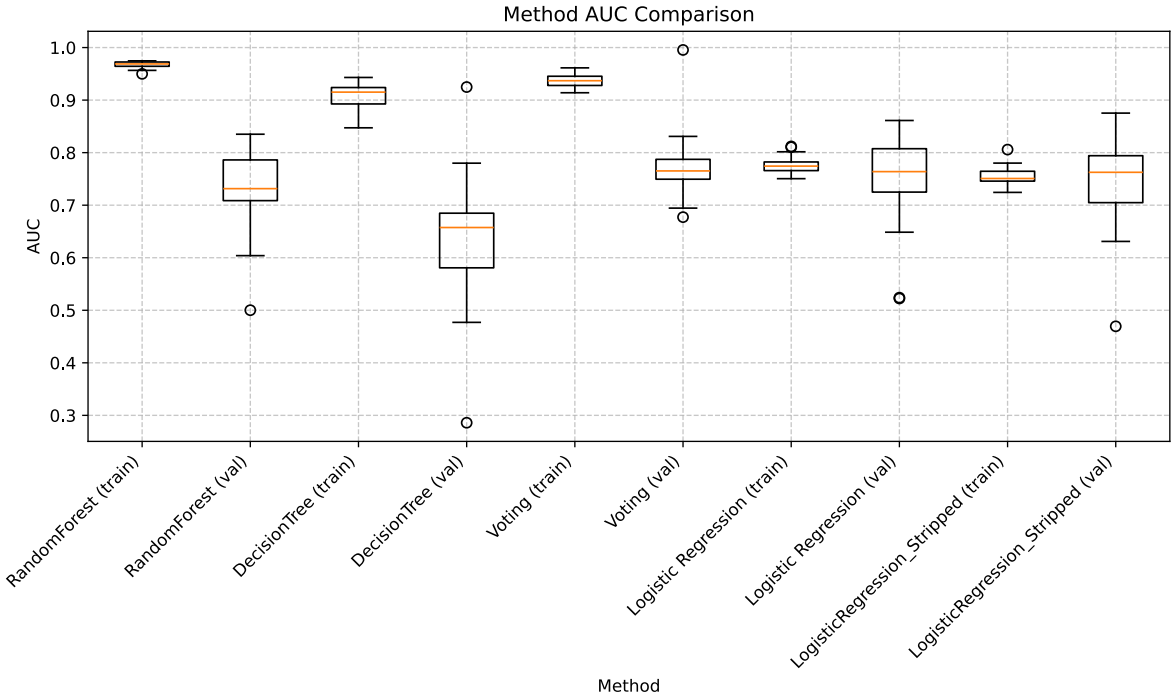

Figure 8: Boxplot of AUC measurements during development phase.

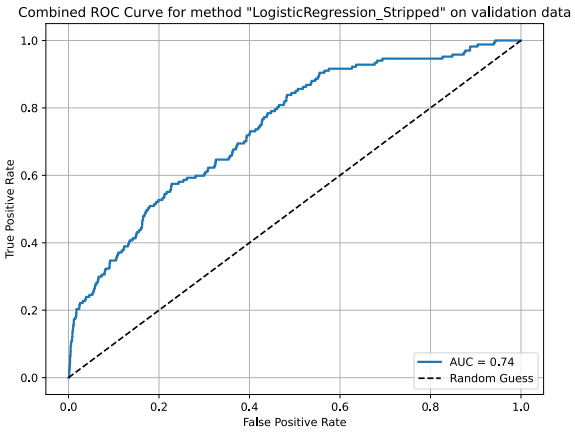

Figure 9: Combined ROC curve for Logistic Regression (without B-feature) on validation data.

a model with a similar algebraic form, such as a logistic regression model, suits the task well.

$$\log\left(\frac{P(y=1 \mid X)}{1-P(y=1 \mid X)}\right) = \beta_0+\beta_1 x_1+\beta_2 x_2+\beta_3 x_3+\beta_4 x_4+\epsilon$$

Under the assumption that a linear model is sufficient for Melanoma classification, it can be argued that a decision tree is a less promising approach because of its inherently non-linear decision boundary. A high tree depth could emulate a linear decision boundary, but would lead to overfitting, which we examined throughout our process.

4.4.2 Better Handling of Class Imbalance

Only 2.33% of our extracted data points are labeled as Melanoma, there being a clear class imbalance. We hypothesize that a KNN approach will suffer from this and limit the choice for the k-parameter to very low values like 1, 2 or 3. This issue arises due to the fact that melanoma points will likely be surrounded by non-melanoma points and the higher the k value the more likely the minority class will be outvoted by the majority class. Unfortunately, low k-values tend to produce complex and volatile decision boundaries that overfit the training data. Logistic Regression only produces linear decision boundaries that are not too complex with only a few features involved. This leads us to believe that Logistic Regression will generalize better and overfit less.

Throughout our development we used weighted logistics regression with the parameter `class_weight = "balanced"`. This assigns weights inversely proportional to class prevalence, which then get used to optimize the log-loss function, hence reducing bias towards the majority class, i.e. non-melanoma in this case.

Similar problems arise when fitting decision

trees on imbalanced datasets, even though we use `class_weight = "balanced"` in our implementation of decision trees, it still performs worse than logistic regression. We believe this is because there are very few melanoma cases, so the tree struggles to find meaningful splits at shallow depths, which translates to poorer performance on unseen data for example the validation or the test set in our case. While class weights do encourage the model to reduce the impurity for minority classes, it still may not be enough to compensate for the low number of melanoma samples and hence exhibit worse performance. This may also be a reason as to why we see in Figure 7 that decision tree on the validation data has a much larger spread compared to training data, indicating the models failure to generalize due to the high imbalance.

5 Results

We present our testing results on the 20% held-out test data and compare recall performance of our baseline and extended method with a statistical testing procedure.

5.1 Baseline Model

The performance of our baseline model on the held-out test data is recorded in Table 1. Where we show the mean and standard deviation for AUC, precision, and recall. These means are taken over 20 grouped shuffles.

Measure	Mean	Standard deviation
AUC	0.4606	0.0882
Precision	0.0196	0.0075
Recall	0.3593	0.1316

Table 1: Baseline model results on test data

We also plot the confusion matrix for the baseline model on the held out test data in Figure 10

5.2 Extended Model

Similar to the baseline model, the performance of our extended model on the held-out test data is recorded in Table 2 and in Figure 11

5.3 Performance Comparison Base - Extended

We investigate whether our extended model performs statistically significantly better in the recall

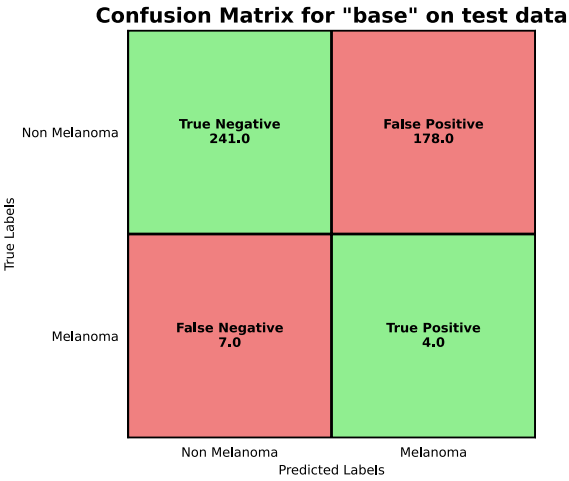

Figure 10: Confusion Matrix for the base model on test data

Measure	Mean	Standard deviation
AUC	0.6374	0.0854
Precision	0.0490	0.0164
Recall	0.5632	0.1559

Table 2: Extended model results on test data

metric than our base model through an appropriate testing procedure. For both methods we resampled the test data set with replacement 20 times to obtain 20 different recall measurements. We use the Jarque-Bera Test and Q-Q plots to verify normality of the data. For the base model we obtain $\chi^2_2 = 1.94$ and $p = 0.370$ and for the extended $\chi^2_2 = 1.09$ and $p = 0.581$. Thus there is no statistically significant evidence against normality in the data, which is confirmed by the Q-Q plots in Figure 12.

We conduct an F test to verify equal variances among both samples and obtain $F_{19,19} = 0.737$ and $p = 0.513$. Under the now verified assumptions of normality and equal variances we conduct a t-test with the null hypothesis

$$H_0 : \ Equal \ Performances$$

and the one-sided alternative hypothesis

$$H_1 : \ The \ extended \ model \ performs \ better$$

The results are:

$$t_{38} = 3.42$$

and

$$p = 0.00075$$

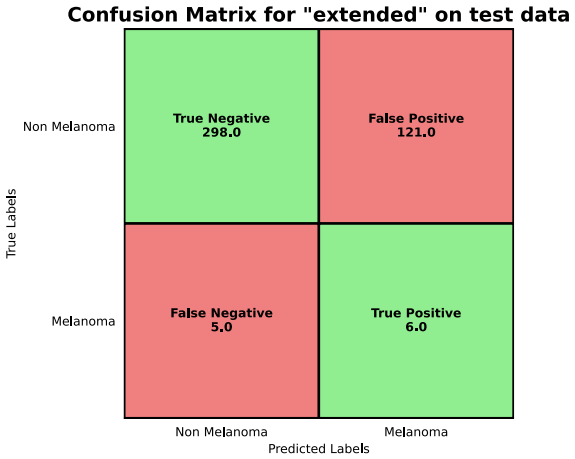

Figure 11: Confusion Matrix for the extended model on test data

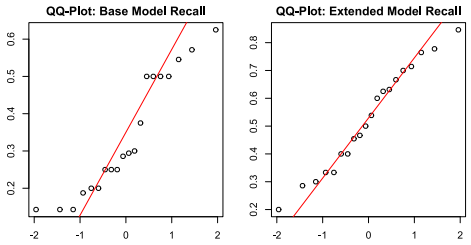

Figure 12: Q-Q plots for recall scores of base and extended model

and thus we can reject the null hypothesis that both methods have equal recall in favor of the alternative hypothesis that the extended model shows better performance at the 99.9% confidence level.

6 Limitations and Obstacles

During the development phase of the model, some issues of various nature arised, posing challenges for the feature extraction aspect, as well as the performance evaluation aspect.
We now discuss what these obstacles are and how we tackle them.

6.1 Limiting Factors for feature extraction

While implementing feature extraction, we faced two main problems linked to the quality and format of the data.

A minor discrepancy regards image sizes.
The functions that needed to apply the mask to the lesion image raised errors due to a 1-pixel size difference between the two.
To avoid this, the involved functions now resize the images to ensure that their dimensions match, making it possible to mask the lesions correctly.

The other roadblock encountered regarded the inconsistent number of connected components (cc) present in a particular mask returned by the designated function (`fCHEESE_extractor()`), and the one that emerged from a visual inspection of the mask itself.
Assuming the main function used, `label()` (part of the module skimage.measure), is accurate and taking into account the fact that the masks are handmade, this is speculated to be due to particularly thin gaps between mask regions that are not visible to the naked eye, likely due to human error.
To control this phenomenon, as mentioned in section 3, we iteratively dilate the mask a different amount of times, computing the number of connected components in each case and returning the mean value.

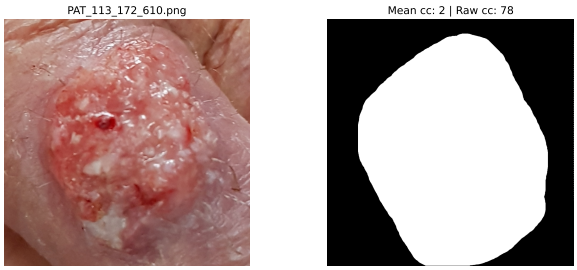

Figure 13: The mean value of cc tends to be closer to the truth, while the raw value can be unpredictable.

6.2 Limiting Factors for Performance

By visual examination of the feature plots for our method, it becomes clear that the current feature distribution makes it challenging to clearly separate the classes from one another Figure 14 Figure 15 Figure 16. Despite this, some features show promising results, such as feature S (”snowflake”) Figure 17 and feature BV (blue veil) Figure 18.
This leads us to believe that not much improvement can be achieved by tweaking the classifier further. Future work should focus on the feature distributions, as there seems to be much room for improvement.

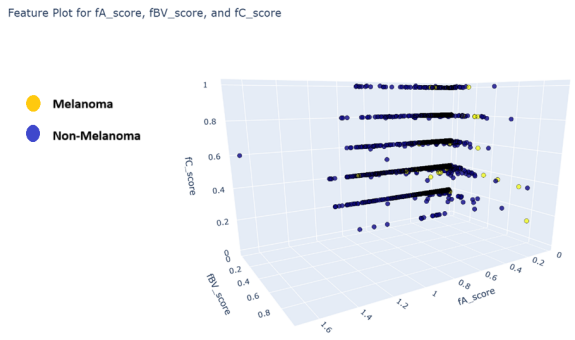

Figure 14: 3D feature plot for features A (asymmetry), BV (blue veil) and C (color).

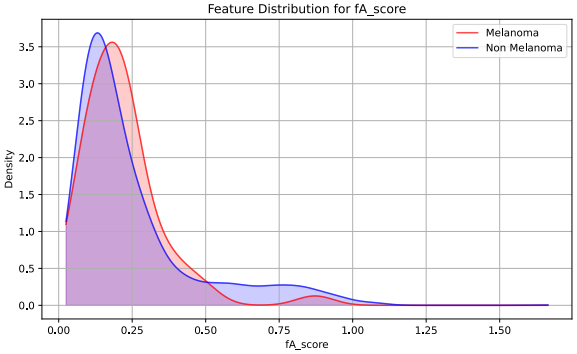

Figure 15: KDE density distribution plot for feature A (asymmetry).

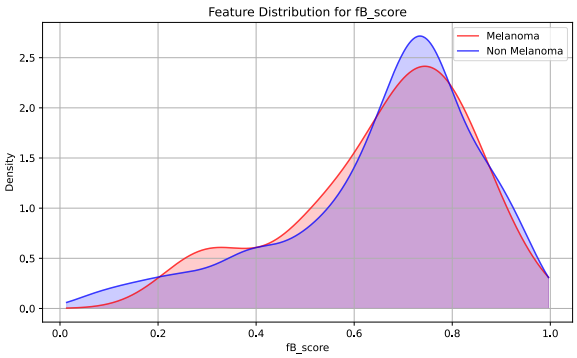

Figure 16: KDE density distribution plot for feature B (border irregularity).

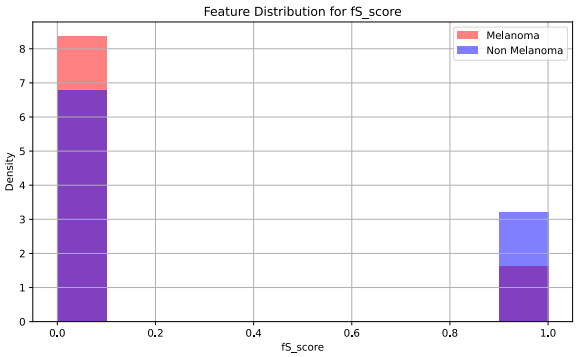

Figure 17: Distribution plot for feature S (”snowflake”).

7 Open Question

One of the big contemporary concerns with data science solutions is their energy consumption. Particularly deep learning and NLP applications have shown to effectuate significant greenhouse gas emissions through their energy consumption (Strubell et al., 2020). As a result, attempts have been made to measure the impacts of such data science approaches (Ligozat et al., 2021) (Thompson et al., 2020).

Not much attention has been devoted to measuring and analyzing the environmental impacts of more traditional data science methods as discussed in (Meulemeester and Martens, 2023). Today, these ”less advanced” applications are used more than ever before across many domains, which motivates us to measure the energy consumption of our own solution. In addition, we try to reduce our power footprint and discuss our results. In the process, we employ the well-known formula of the ”ABCD Rule” (Nachbar et al., 1994) to avoid repeatedly fitting and evaluating classifiers. Finally, we draw a possible scenario for which we extrapolate energy savings and reflect on our results.

7.1 Previous Research

In the 1990s, Wilhelm Stolz and his team developed the ABCD rule(Nachbar et al., 1994). This method gives the doctors a structured way to check if a skin lesion is melanoma by looking at four main features: Asymetry, Border, Color, and Dermoscopic structures.

$$y = 1.3a + 0.1b + 0.5c + 0.5d$$

Where the A-feature can take values 0 to 2, the B-feature can take values 0 to 8, the C-feature takes values takes values 1 to 6 and the D-feature takes values 0 to 5. A practitioner would then classify any lesion exceeding

$$y = 4.75$$

as melanocytic cancer. Since the project relies on smartphone images rather than dermoscopic images, we adapted the original ABCD rule. While we have preserved the A, B, and C components

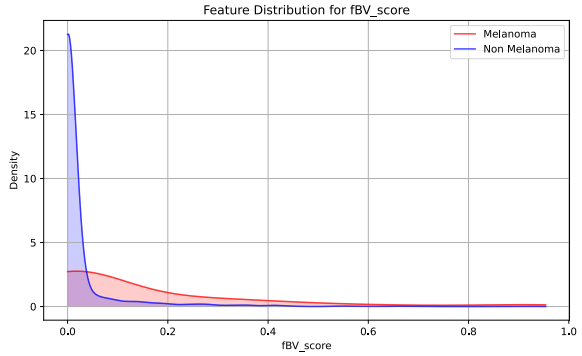

Figure 18: KDE density distribution plot for feature BV (blue veil).

we replaced the Dermoscopic structures with visible symptoms such as evolution, size and patient reported changes. This makes the algorithm more suitable for real world use where dermoscopes may not available.
We also consulted with a medical professional before making this change, to ensure that our adjusted method still made clinical sense.

7.2 Adapted Features

In this paragraph we present concretely how we translate these visual features into code functions, diving into the main challenges, the necessary changes and the techniques used to implement them.

7.2.1 A - Asymmetry

The A feature in the open question measures the asymmetry of the skin lesion. It differs slightly from the A feature implemented in the baseline model. The main difference is that in the baseline model, feature A returns continuous values from 0 to 1, whereas in this implementation it returns discrete values, namely 0, 1 and 2. In the open question implementation feature A looks at each axis separately, value 1 is assigned if there is asymmetry in one of the axes, value 2 is assigned if there is asymmetry in both axes, else value 0 is assigned. Feature A here uses a threshold defaulted to 0.2, meaning if more than 20% of the total pixels in that axis are mismatched pixels, then there is asymmetry in that axis. Recall baseline feature A, which returns a ratio of the mismatched pixels in the mask over the total pixels, and then takes the average of that score in each rotation.

In reality, Stolz formula asymmetry does not only concern the shape but also the color. However the color asymmetry is not accounted for in our implementation, primarily due to time constraints, but also because shape alone gives a nice estimate.

7.2.2 B - Border Irregularity

The B feature measures the ”Border Irregularity” on a scale from 0 to 8. +1 is added to the score for each 45 ° sector of the lesion showing a well-defined border(Nachbar et al., 1994). Our implementation converts the lesion pictures to grayscale and samples 10 radial lines per 45 ° degree sector from the lesion center towards the image borders. We assign each sector a score that is the mean of the maximum gradient values along the 10 radial lines in that area. This is based on the assumption that sectors with a well-defined border tend to have higher maximum gradients from the center towards the edges than those with fading borders do. Finally a K-Means algorithm is used to divide the sectors into two clusters based on their gradient scores. This dynamic approach is necessary because various challenges, such as inconsistent lighting, lead to very different gradient score ranges across images.

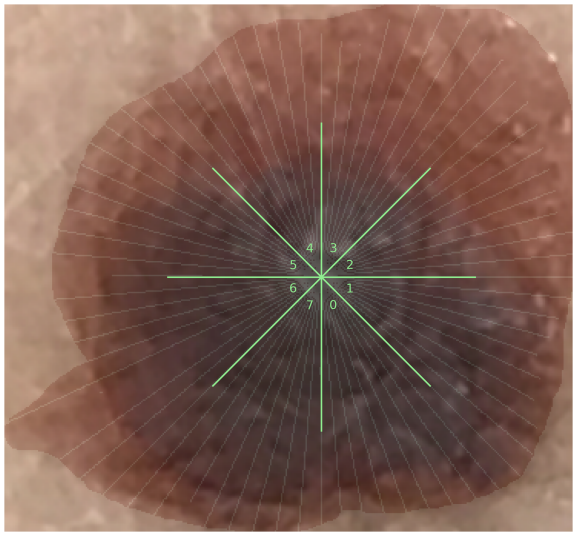

Figure 19: Visualization of the computation of the adapted feature B (border irregularity).

7.2.3 C - Color

The changes influential to the adaptation of the color feature, bring about a more precise way of determining the list of hues present in a lesion image. Instead of collecting the 6 most prevalent yet diverse colors, it will look for a specific 6-tuple of them, defined by an RGB range with the usage of if statements. The return will remain the same, the

length of this list, but the approach of collecting these colors has been somewhat shifted to adapt to our formula.

7.2.4 D - Dermoscopic Structures

The D feature originally was a key feature to recognize the Melanoma, by looking for different structures in the pictures, like dotes, globes, networks, branched streaks, and structureless areas. However, we cannot disregard the fact that these pictures are not made with dermoscopes. So we consulted with a medical professional about it, and we have choose other parameters, that could be useful for the prediction, with weights, such as the `age` 0.8, `size` 1.2, `evolution` 1, which is just the union of the `grew` and `changed`, `itch` 0.1, `hurt` 0.1, `elevated` 0.2, and the `bleed` 0.8 from the `metadata.csv`. (Iványi, 2006) (Boulos Mansour, Michele Donati,) This way the D feature scales from 0 to 5.

7.3 Evaluation of Adapted Features

Having adapted the needed features to fit coding constraints and to accomodate the quality and range of the available data, we consulted a doctor to visually annotate the values corresponding to the A, B, and C features for 27 of the images present in our data set.
Note that the D feature has not been included in the table, since the standard D feature differs quite significantly from our own version, making the two not comparable.

7.3.1 Comparison of Features and Manual Annotations

This new set of measurements is used to test our coding approach to the formula, originally meant to be applied via visual inspection of the lesion.
In particular, we compute the values for the features in question for the 27 selected images using our model, then, for each of them, we check the accuracy of the predictions by comparing them with the manual annotations.
The results of this can be consulted in the Table below, where the computed values are colored accordingly to their accuracy with respect to the manual annotations: a green cell means they are equal, a yellow cell means the computed value is off by 1, an orange cell means it is off by 2, and for greater differences the cell is red [Figure 20].

The resulting accuracy is 74.07% for feature A, 3.70% for feature B, and 18.52% for feature C.

As we can see, the function responsible for extracting feature A is mostly reliable. Unfortunately, the function linked to feature B is off by at least 2 from the actual value of the feature on 9 occasions, and is correct one single time. The function used to extract feature C is discretely accurate, since it does produce 11 instances in which the computed feature is off only by 1 from the actual value and, broadly, it is close to the optimal results.
It is also worth underlining how comparing visual evaluations with code output is not simple, and small differences between the two are to be expected and do not impact the final outcome significantly.
In general, the implementation of the functions needed to adapt the formula for this code-based approach is tedious and definitely not straightforward. Furthermore, the quality of the images in the data set is not up to the standards held for pictures on which the formula is commonly applied on.
Nonetheless, when applying the Formula classifier to a random sample of images from the data set, the results appear to be discrete, as we will see in Section 7.3.2.

7.3.2 Performance of the Formula classifier

To evaluate the Formula classifier, we run the method over all valid image-mask pairs in the data set (2100 pairs). In this specific case we do not take into account the AUC value, since the decision threshold cannot vary, and we do not have the necessity to train the classifier.
We therefore obtain values for both recall and precision, and a confusion matrix, which can be consulted below [Figure 21].
The precision turns out to be 3.11%, while the recall is 81.60%.
This translates into a substantial amount of additional medical checks for people without melanoma, but few cases in which melanomas will not be detected as such.
It is worth noting that all images in the data set do not contain heathy skin, and the formula is based on characteristics present also in other skin diseases, resulting in some overlap. It is therefore harsh to label as false positives unhealthy skin images identified as melanoma. Refer to Section 4.2.2 and Section 4.2.1 for a more in-depth definition and practical meaning of these performance measurements.

	Feature A predicted	Feature A actual	Feature B predicted	Feature B actual	Feature C predicted	Feature C actual
PAT_56_86_479.png	2	2	3	5	5	2
PAT_59_46_537.png	2	2	3	5	4	2
PAT_70_107_591.png	2	2	5	0	2	3
PAT_109_868_113.png	2	2	1	0	3	2
PAT_320_681_410.png	0	2	3	0	2	2
PAT_324_1465_43.png	2	2	2	0	3	2
PAT_340_714_68.png	2	2	5	6	6	3
PAT_471_909_394.png	2	2	2	4	4	3
PAT_490_933_17.png	2	1	4	6	2	1
PAT_627_1188_503.png	2	2	4	0	5	3
PAT_656_1246_483.png	1	2	6	4	5	2
PAT_680_1289_585.png	2	2	2	4	3	2
PAT_754_1429_380.png	2	2	3	5	4	3
PAT_795_1508_925.png	2	2	3	0	4	1
PAT_884_1683_538.png	2	2	3	2	6	4
PAT_895_1699_872.png	2	2	3	2	3	3
PAT_966_1825_584.png	2	2	2	2	3	3
PAT_995_1867_165.png	2	2	3	0	3	3
PAT_1113_458_387.png	0	1	5	6	6	2
PAT_1259_892_793.png	2	2	2	3	3	4
PAT_1286_1000_517.png	2	1	3	8	5	2
PAT_1420_1460_951.png	2	2	7	6	2	3
PAT_1653_2916_346.png	2	2	4	2	2	3
PAT_1698_3122_83.png	1	1	6	8	2	1
PAT_1928_3876_437.png	1	2	1	8	4	3
PAT_2017_4164_500.png	0	2	4	0	4	4
PAT_2103_4581_72.png	2	2	4	8	5	2

Figure 20: Table that illustrates computed vs actual values for A, B, C features of 27 images.

While performance is not the primary goal of this open question section, because of the lack of an appropriate data set, as well as reliable feature-extraction functions, a high recall testifies the potential of the Formula classifier. Time and energy alike are saved, by using a training-free model such as the one presented.

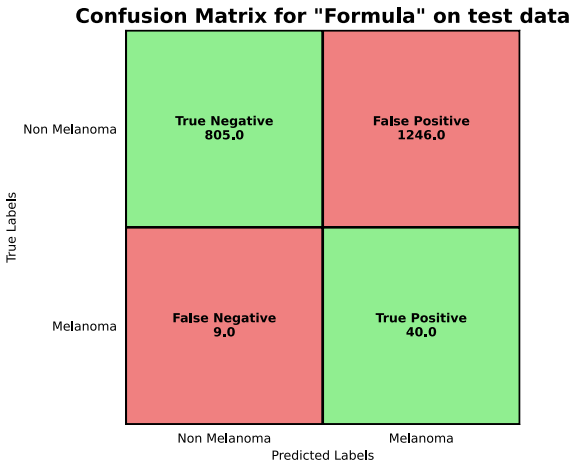

Figure 21: Averaged Confusion Matrix for Formula Classifier.

7.3.3 Consultant's Statement - Dr. Tézsla Zs.

"At the start of the project, I was invited by Gergely Takács for a consultation, the topic of which was to review the fundamental macroscopic characteristics of melanoma (malignant pigment cell tumor). During our discussions, we talked about the well-known "ABCDE" rule, the epidemiology and etiology of melanoma, and we also touched on a clinically significant and common subtype with atypical morphology: nodular melanoma. We partially covered the basics of the dermatoscopic scoring system as well.
Later on, professional assistance was requested to evaluate the images used as samples.
Following the development of the "formula," we compared the images I had scored with their results. The comparison showed that there were only minor discrepancies in the assessment of asymmetry and color, while the evaluation of borders still required further refinement. The larger differences were mainly traced back to the inaccuracy of certain images used as "masks", and refining these could significantly improve the formula's effectiveness and provide relevant assistance in everyday clinical practice.

The images in the database require considerable selection (many are blurry, shadowed, or taken from unfavorable angles, which greatly hinders the successful processing of relevant data).
Finally, I would like to highlight that, in my daily work in pathology, I regularly perform macroscopic descriptions of skin lesions that have been removed, which serves as an important foundation for later microscopic/histological diagnosis. However, I must emphasize that I do not hold a dermatology specialization."

7.4 Possible Energy Savings

We present our energy measurements both for the regular approach and our proposed open question approach. Based on this we try to extrapolate our electricity savings to a possible research scenario and conclude on our results.

7.4.1 Measurements

The feature extractions for all of our features are monitored for the total batch of images. Classifier training and evaluation occurs repeatedly for the first method, but only one full test run is needed for the second method. Our results of the total energy usage for each of the two methods are were recorded in kWh, and, as such, are ready for comparison and conclusion, thus pushing towards a resolution to our open question.

7.4.2 Possible Research Scenario Extrapolation

We hypothesize a possible research project and extrapolate our energy measurements to this hypothetical scenario to extrapolate possible energy savings.

Hypothesized Scenario
Suppose a team of researchers were developing a computer vision method for computer aided melanoma diagnosis. Suppose their dataset consists of 1 Million images labeled as either 1 (Melanoma) or 0 (non Melanoma).

Method 1: Classical Data Science Approach
Under the assumption that the team of researchers uses the same features as used in our extended method (A - asymmetry, B - border irregularity, C - color, BV - blue veil, S - "snowflake", Ch - connected components), we can extrapolate the energy cost of feature extraction for 1 Million images from our measurements. We assume that $EF \sim n$ where n is the number of images and EF is the e Under the linear assumption that $EF \sim n$ we estimate:

$$EF_{h,1} \approx EF_o * \frac{n_{h,1}}{n_o}$$

where EF is the energy consumed by feature extraction, n is the number of images in the dataset and the subscripts o and h,1 refer to "our scenario" and "hypothesized scenario 1" respectively. Therefore:

$$EF_h \approx 0.044\,kWh * \frac{1,000,000}{2298} = 19.15\,kWh$$

In addition we assume that $EC \approx n$ and estimate:

$$EC_{h,1} \approx EC_o * \frac{n_{h,1}}{n_o}$$

where EC is the energy consumed by repeatedly fitting and evaluating the classifier. We estimate that classifiers are fitted and evaluated on the data 5000 times throughout the research project. Each measurement is conducted over 20 random splits, amounting to 100,000 fit and prediction runs of classifiers. Therefore:

$$EC_h \approx 0.16\,kWh * \frac{1,000,000}{2298} = 69.63\,kWh$$

We sum to obtain an estimate for the total energy consumed by Scenario 1:

$$ET_{h,1} = EF_{h,1} + EC_{h,1} = 88.78\,kWh$$

Method 2: Fixed Formula Approach with Feature Evaluation by a Medical Professional
In this approach we assume that the team of researchers has access to a medical professional to repeatedly assess the quality of developed features. Features are developed and evaluated on small random batches of the dataset to avoid the energy cost associated with extracting features on all images. This process is continued until an agreement within debatable range between extracted features and professional annotations is reached. Instead of fitting a classifier, the fixed formula is applied to predict labels for the images. To obtain a conservative estimate of energy saving we estimate

$$EF_{h,2} \approx EF_{h,1}$$

although it is likely much less. Therefore:

$$EF_{h,2} \approx 19.15\,kWh$$

This scenario consumes no energy for fitting and evaluating the classifier repeatedly because only

one test run is conducted. We regard the energy cost associated with this one test run as negligible because it is approximately $1 * 10^{-5}\, kWh$ and therefore estimate $EC_{h,2} \approx 0\, kWh$. We sum to obtain an estimate for the total energy consumed by Scenario 2:

$$ET_{h,2} = EF_{h,2} + EC_{h,2} = EF_{h,2} =$$

$$= EF_{h,1} \approx 19.15\, kWh$$

Energy savings:
We estimate the energy saved as:

$$ES \approx ET_{h,1} - ET_{h,2} = 69.63\, kWh$$

This is equivalent to driving the Mini Cooper E electric vehicle for about 475 km under average conditions(EV Database, 2023) or running about 77 washing machines (900 watts, 1 h each wash).(Energysage, 2024).

7.4.3 Conclusions and Possibilities for Improvement

Our open question highlights the potential for energy savings when transitioning from a traditional, computation-heavy data science workflow to a fixed formula approach supported by expert input. We have estimated a reduction in energy consumption from approximately 88.78 kWh to 19.15 kWh - a savings of 69.63 kWh. This result demonstrates that integrating domain expertise in the feature evaluation process can significantly reduce computational demand while maintaining diagnostic effectiveness 7.3.2. Although we used a conservative estimate for feature extraction energy, the actual consumption in the fixed formula approach is likely even lower, since only small batches are processed during development.
Further improvements could include refining the formula to reduce reliance on computationally heavy features and reviewing the dataset to eliminate low-quality or re-appearing images, which currently introduce inefficiencies and noise.
Future research could explore the refined masks which would improve the overall performance, or hybrid methods that combine limited training with expert-reviewed features.

8 Conclusion

8.1 Improvments

In this section we discuss some of the limitations with both our model and the feature extraction and how we would improve this in future projects.

Firstly we would like to discuss the hair feature, as mentioned in section 3, the hair removal feature was not implemented in the final dataset, instead it was limited to just labeling the images. This is definitely something we would like to improve. Had we had a reliable hair feature which does not tend to falsely identify skin segments as hair; we might have been able to obtain better results in the other features, especially the ones which use the image rather than the mask.

Secondly feature A, which for the most part was very stable. We would like to make improvements to the function responsible for extracting feature A in the open question to incorporate color into the asymmetry score. This would further align our implementation with the Stolz formula, which may further improve the performance of the method.

Another area in which we think we could improve is feature diagnostics, namely understanding why certain features return unexpected values. As an example, feature A momentarily returned values above 1 in the baseline model, which we fixed by making sure values are within range. Comparing our feature extracted values with our own logic or existing values from dermoscopic images, might improve separability and hence model performance.

Lastly we would like to mention how incorporating the metadata columns into the model would effect performance. As an example, how adding the body part on which the lesion grew to the model might improve its performance. Here we also stress the need for a more balanced dataset. Nonetheless, we are aware that in medical datasets it is common to have less samples of rare diseases such as melanoma. We do believe that having more melanoma samples would have improved our model.

8.2 Final Remarks

In Conclusion, we saw that smartphone images do give some promising results for melanoma detection, although the ones provided did not excel in quality, as stated in Section 7.3.3. Moreover, at times we also realized its limitations compared to dermoscopic images.

Acknowledgment

We would like to express our sincere gratitude to Dr. Tézsla Zs., resident pathologist at the Pathology Department of Ödön Jávorszky Hospi-

tal, Hungary, for her valuable consultancy related to feature extraction, the formula used in our open question and for her annotations for our adapted features.

References

[ACS, 2023] ACS, A. C. S. (2023). Tests for melanoma skin cancer. Accessed: 2025-05-29.

[Boulos Mansour, Michele Donati,] Boulos Mansour, Michele Donati. Pathologyoutlines.com website, invasive melanoma. https://www.pathologyoutlines.com/topic/skintumormelanocyticmelanoma.html URL accessed May 26th, 2025.

[Energysage, 2024] Energysage (2024). Washing machine energy consumption. https://www.energysage.com/electricity/house-watts/how-many-watts-does-a-washing-machine-use/. Accessed: 2025-05-27.

[EV Database, 2023] EV Database (2023). Mini cooper e electric. https://ev-database.org/car/1949/MINI-Cooper-E. Accessed: 2025-05-27.

[Iványi, 2006] Iványi, A. (2006). *Bőrpatalógia*. Medicina Könyvkiadó Kft.

[Ligozat et al., 2021] Ligozat, A.-L., Lefèvre, J., Bugeau, A., and Combaz, J. (2021). Unraveling the hidden environmental impacts of ai solutions for environment. *arXiv preprint arXiv:2110.11822*.

[Meulemeester and Martens, 2023] Meulemeester, B. and Martens, D. (2023). How sustainable is "common" data science in terms of power consumption? *Sustainable Computing: Informatics and Systems*, 38:100864.

[Mostame, 2023] Mostame, P. (2023). Hair removal from skin images. https://www.kaggle.com/code/parhammostame/hair-removal-from-skin-images/notebook URL accessed May 27th, 2025.

[Nachbar et al., 1994] Nachbar, F., Stolz, W., Merkle, T., Cognetta, A. B., Vogt, T., Landthaler, M., Bilek, P., Braun-Falco, O., and Plewig, G. (1994). The abcd rule of dermatoscopy. high prospective value in the diagnosis of doubtful melanocytic skin lesions. *Journal of the American Academy of Dermatology*, 30(4):551–559.

[Pacheco et al., 2020] Pacheco, A. G., Lima, G. R., Salomão, A. S., Krohling, B., Biral, I. P., de Angelo, G. G., Alves Jr, F. C., Esgario, J. G., Simora, A. C., Castro, P. B., Rodrigues, F. B., Frasson, P. H., Krohling, R. A., Knidel, H., Santos, M. C., do Espírito Santo, R. B., Macedo, T. L., Canuto, T. R., and de Barros, L. F. (2020). Pad-ufes-20: A skin lesion dataset composed of patient data and clinical images collected from smartphones. *Data in Brief*, 32:106221.

[Strubell et al., 2020] Strubell, E., Ganesh, A., and McCallum, A. (2020). Energy and policy considerations for modern deep learning research. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13693–13696.

[Thompson et al., 2020] Thompson, N. C., Greenewald, K., Lee, K., Manso, G. F., et al. (2020). The computational limits of deep learning. *arXiv preprint arXiv:2007.05558*, 10.

[Warjurkar and Ridhorkar, 2021] Warjurkar, S. and Ridhorkar, S. (2021). A study on brain tumor and parkinson's disease diagnosis and detection using deep learning. In *Proceedings of the 3rd International Conference on Integrated Intelligent Computing Communication Security (ICIIC 2021)*, pages 356–364. Atlantis Press.